Image Segmentation for Color Document Analysis※

Yi-Sheng Lin (林怡聖) and Dr. Wen-Hsiang Tsai (蔡文祥)

Institute of Computer and Information Science
Department of Computer and Information Science
National Chiao Tung University, HsinChu, Taiwan, Republic of China

Abstract

An approach to segmenting and recognizing color document components including texts and graphics is proposed. First, a mean-cut algorithm, which is a revised version of the median-cut method, is used to quantize a color image and decompose the result into a set of planes, each with a single color. By the mean-cut algorithm, a number of reduced colors can be decided appropriately. A multi-spectral constrained run-length algorithm is proposed to extract objects from the color planes. The algorithm can extract texts and merge non-texts of each plane to form graphic components. Even when a textline resides on a complicated background, it still can be taken out easily. Finally, a new method is also developed to recognize textline components. This method can eliminate the shortcoming of using statistical features in textline recognition. Some experimental results are also shown to prove the feasibility and practicability of the proposed approach.

1. Introduction

1.1 Survey of Related Studies

A document image is like a set of raw data which are meaningless for computers. These raw data need be analyzed and transformed into an equivalent representation which can be apprehended by computers. All of the information should be preserved in an efficient form.

The quality of a document image is related to the scanning or digitization system. It is very important to choose a suitable resolution because the analysis results are influenced by the resolution strongly. Generally speaking, the selection of the resolution depends on the character size. A resolution about 200 to 250 dpi is suggested for character recognition. This resolution is commonly used in office document digitization. As a result of the high resolution, enormous storage is required and the resulting analysis work is more complex and heavy, especially for color document analysis.

The techniques for document analysis have been studied for many years. Wong and Casey started to design and build an experimental document analysis system [1]. Some of the other existing document analysis systems [2-4] were designed to deal with publications that have a specific format. Such a kind of approach is called as a top-down method. It can deal with documents rapidly. However, it is not powerful enough. Another approach opposite to the top-down method is bottom-up, which analyzes documents with arbitrary layout. It merges related pixels to constitute text lines or graphics. In fact, a pixel carries little information. The local information from the neighboring pixels should be extracted. You can imagine a microbe residing on a newspaper. It cannot recognize a word unless it arises to look down at the newspaper. The constrained run length algorithm proposed in [5] can be said to be based on this principle.

A general document analysis system includes several basic steps. The first step is image preprocessing, which improves images with poor quality resulting from inappropriate operations or inapt devices. With preprocessing, the recognition rate can be increased in the later steps.

Thresholding is often used in the preprocessing stage. When dealing with a gray-scale image, it should be binarized into the bitmaps at first. But this may result in a low recognition rate if an improper thresholding value is chosen. Local thresholding [6] or optimal thresholding [7] is more applicable to image refinement.

The second step is block extraction and type recognition. In this stage, an image is decomposed into independent blocks, which may overlap one another. Each block may contain text lines, halftone images, or graphics. The smearing algorithm [5, 8-9] has been widely used in block extraction. It will be reviewed in Section 3. Another segmentation approach often used is to utilize horizontal and vertical projection profiles [10-11] which are also called as XY cuts. This scheme is suitable only for documents with simple layouts. A third segmentation algorithm is called SPACE [12] which is based on a peak-pixel method for detecting the halftone graphics and on edge detection performed on a continuous black and white representation of the image. The Hough transform can be also used to detect the structures of the form documents [13]. The closed contours of the connected components are approximated by piecewise linear line segments. Jain and Bhattacharjee proposed a multichannel filtering-based texture segmentation method which can be used for text-graphics separation [14]. Two-dimensional Gabor filters are used to compute texture features. In fact, the contour following technique is the simplest method for document segmentation [15]. However, it might spend much time. Recently, the neural network is used for optical character recognition (OCR), and it can be used for the segmentation of document images as well [16].

※ This work was supported partially by National Science Council, Republic of China under Grant NSC 83-0408-E-009-010.

As soon as a block is extracted, the recognition of the block type is performed. If a block is recognized as a text line, it need be partitioned further into individual character blocks. These characters are then recognized in the optical character recognition (OCR) stage. OCR has been studied for many years. It is often excluded from the field of document analysis and belongs to another research field. In our study, we do not emphasize too much in OCR.

As to document understanding, which is less stressed, it extracts the logical relationships between the textlines, graphics and photographs. The relationships mentioned here are the sequence in which a document is read. By virtue of document understanding, the information can be stored in an efficient way. And the retrieval of document contents can be expedited. A logical tree representation is a good approach [17].

1.2 Proposed Approach

1.2.1 Overview of Proposed Approaches

In our system, the first stage is color processing. Color is an important information. But it will put heavy burden if a true color image is processed. In fact, 256 or less colors are enough. In this study, we try to reduce the number of colors. An adaptive mean-cut color quantization algorithm is designed. By analyzing the color distribution, an appropriate number of colors can be determined.

As for object extraction, to reduce storage space and processing time, an algorithm called the multi-spectral constrained run length algorithm (MCRLA) is designed. This is extended from the CRLA used in gray-scaled document analysis [5]. A color image is quantized into less than 256 colors in advance. Consequently, 256 color planes are produced. With the MCRLA, the objects of each color can be extracted concurrently.

After objects are extracted, the object types need be recognized. In order to analyze the documents with texts residing on a complicated background, such as graphics, a more precise recognition scheme is necessary using more subtle features in addition to the conventional statistical features. According to the property of the text that is constituted with slender lines, we propose a textline detection method to recognize texts of any scaling.

1.2.2 Assumptions

In order to reduce the complexity of the problem, some restrictions and assumptions are made in this study. In spite of these restrictions, the system is still applicable to the majority of color documents.

1. In our study, we try to recognize the text and graphic components. The recognition for tables and charts is not discussed.
2. A character is assumed to have uniform color. Texts with gradient colors are avoided.
3. The documents contain either horizontal or vertical text lines, but not both.

2. Color Quantization Techniques

2.1 Review of Recursive Median-cut Color Quantization

Color quantization techniques can be viewed as an application of clustering. Lots of clustering schemes have been proposed. But most of them are not suitable for color quantization. Some of them spends too much time in analysis and some produces quantization whose quality is not good enough. A so-called median-cut color quantization method which are usually adopted in commercial software is reviewed as follows.

In the median-cut color quantization method, a qunatizer is designed to partition the color space into quantization cubes as shown in Figure 1(a). Each of the cubes is assigned a representative color, which can be the mean of the 3-tupled vectors in the cube. And all the colors within a cube are reproduced by the representative value.

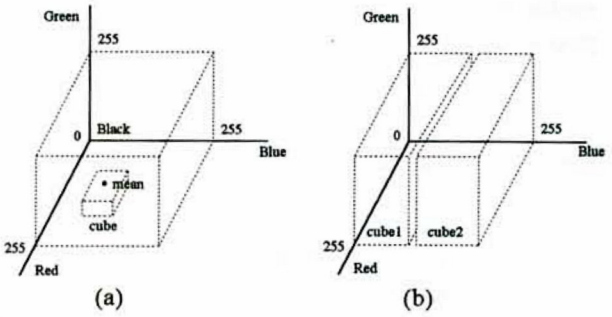


Figure 1. (a) A quantized cube associated with a representative value. (b) The original color space is divided into two cubes at first.

Initially, the entire color space is regarded as a single cube whose representative color is the mean vector of the r, g, and b components. The histogram for each color component can be created and the variance for each histogram is computed by

$$\begin{aligned} var(R) &= \sum_{r=inf}^{sup} p(r) \cdot |r - mean\_r|; \\ var(G) &= \sum_{g=inf}^{sup} p(g) \cdot |g - mean\_g|; \\ var(B) &= \sum_{b=inf}^{sup} p(b) \cdot |b - mean\_b|; \end{aligned} \tag{1}$$

The distribution with the largest variance is split into two parts according to the median-cut method. The median-cut method is similar to median filtering. The median m of a set of values is such that half the values in the set are less than m and half greater than m. It is necessary to sort the values. Take the set, {7, 3, 6, 3, 1, 6, 8, 4, 7}, as an example. They can be sorted as {1, 3, 3, 4, 6, 6, 7, 7, 8} and the median is the 5th largest value, which is 6.

As to color quantization, the cut point can be determined in similar ways. After the partitioning, a cube is divided into two smaller ones, like Figure 1(b). Continuing the partitioning on the smaller cubes, all the color coordinates with similar values will cluster together. The representative colors become

more and more precise and the variance for each cube is reduced. 2^n cubes will be always be created because half of the colors are classified into a cube and the other half are in the other cube. The splitting process is just like building a binary tree.

2.2 Mean-Cut Method for Color Quantization

In our study, we try to use mean-cut instead of median-cut. The results are still good, or even better. In fact, mean-cut is usually adopted in thresholding and performs very well. It is also suitable for color quantization.

The splitting principle of the median cut method is also revised in our approach. For each splitting operation, a cube with the largest variance is selected to be split. An appropriate threshold value T is preselected in order to measure the quality of the quantized image: If the variance of the cube is smaller than threshold T, it never be split any more. The proposed quantization algorithm is as follows.

Mean-cut Quantization Algorithm:

Step 1: Input a color image.
Step 2: Compute the means and variances of the R, G, and B components individually for the entire color coordinate space which is the only cube so far.
Step 3: Find the cube with the largest variance among the existing cubes.
Step 4: If the largest variance is less than the threshold value T, then return all the cube information and exit.
Step 5: Split the cube into two subcubes with the mean of the cube as the cut point.
Step 6: Compute the means and variances of the R, G, and B components for the two new subcubes.
Step 7: Go to Step 3.

3. Image Segmentation

3.1 Review of Constrained Run-Length Algorithm (CRLA)

In monochrome document analysis, a gray-scale image is binarized at first into a bitmap which is scattered with black and white pixels. Most of the documents are assumed to have a common layout model including isolated headlines, textlines, graphics, etc. Evidently, the image segmentation problem can be transformed into the problem of block extraction.

The principle of the CRLA is to smear a binarized image by connecting the black pixels which are close to each other. Let 1 denote black pixels and 0 denote white ones. Given an arbitrary sequence of 0's and 1's, and a threshold value, say TH, which determines whether the black pixels are located closely enough or not, the white pixels between two black pixels whose distance is smaller than or equal to TH are replaced by 1. For example, if TH = 3, the sequence

1 1 0 0 1 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 1 1 1

will be transformed into the following sequence by the CRLA

1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1.

Smearing the binarized images vertically and horizontally will produce two distinct intermediate bitmaps, say M1 and M2. Combining bitmaps M1 and M2 with an AND logical operation, the resulting image is almost desired. The textline blocks and graphic blocks are usually isolated so that each component can be extracted.

The CRLA connects elements by padding 1 between two close black pixels. The resulting image will be then divided into blocks each of which is the minimum inscribing rectangle of an extracted 8-connected component object. Based on the CRLA and the above 8-connected component extraction method, a new algorithm of region growing and segmentation of multi-spectral bands is proposed in the next section.

3.2 Proposed Multi-spectral Constrained Run Length Algorithm

Human eyes are able to distinguish texts from background because text colors differ from neighboring pixels. This gives us a good idea. According to the assumption that a text is constituted by pixels with an identical color, the text can be detected by considering only those pixels with a specified color.

Unfortunately, this scheme causes another problem. A true color image can display about 1,600,000 colors. It puts a heavy burden if we analyze the color image for each spectrum. In fact, many computer monitors cannot display so many colors. They usually quantize true colors into 256 colors or less, and the quality is still good. For this reason, color quantization can be utilized to eliminate this trouble. In the previous section, an algorithm of color quantization has been proposed. Using this color quantization method to decompose a color image, multiple bit planes of different color spectra can be produced, as illustrated in Figure 2.

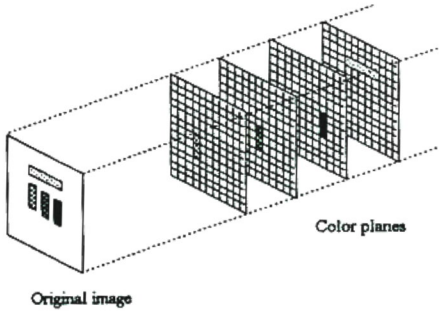


Figure 2. An image is decomposed into bit planes using colors.

These bit planes can be viewed as a sequence of gray-scale images and smeared by the CRLA one after another. But this is not efficient. The more spectra image possesses, the more passes the CRLA needs. With today computers, it will spend much of time.

On the contrary, if these spectra planes are smeared at the same time, the time-consuming problem will be solved but many of the image bitmaps need be kept on the memories. It is also another trade-off.

A run-length image representation is proved to be suitable [17]. It is more efficient than a bit-map representation. With the run-length technique, a multi-spectral constrained

run length algorithm (MCRLA) which can be applied parallelly to all the color planes is proposed in this study. The details are described in the following.

Initially, the original image is quantized into an index plane. Each pixel on the index plane is labeled with a number which denotes the color index or color spectrum. The extracting procedure is performed for the pixels with respect to each spectrum, just like processing all color planes individually. Evidently, the space consuming problem is therefore solved.

The MCRLA performs horizontal scanning line by line. It is an iterative procedure including run collection and block growing. In the iterative process, two tables are required. One is to register collected runs after each scanning and the other is to record the existing block embryos which are growing up. The margins of the runs and blocks are recorded in the tables.

Various spectra for each pixel are encountered during the scanning. The adjacent pixels with the same spectrum are smeared to a run. It is like the CRLA procedure but only the left and right margins of the runs need be recorded in the table. It is possible that runs of different spectra overlap one another. There is no ambiguity, however. An example is shown in Figure 3.

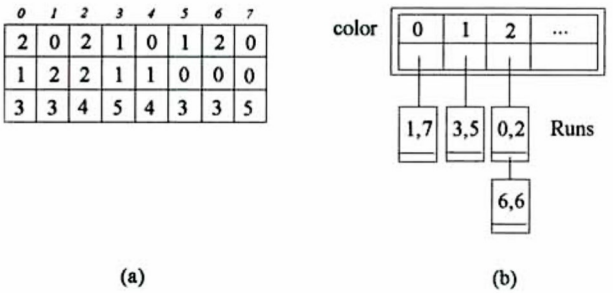


Figure 3. An example to illustrate the run collecting process with smearing threshold THh=3. (a) A color index plane; (b) the runs collected in the table after the first horizontal scanning. Each run records the left and right margins.

After collecting runs with different spectra in each scanning, each run is checked to see whether it is adjacent to any of the corresponding block embryos with the same spectra in the table. This procedure is like creating connected components. Those block embryos adjacent to the runs will be updated as shown in Figure 4(a). If a block is not updated in a certain scanning, it is regarded as a connected component and will be recognized immediately.

In the block growing process, a run may be adjacent to several block embryos with identical spectra. These blocks should be merged into a single block as shown in Figure 4(b).

After a block is determined, the pixels with a specified spectrum in this block area is selected and the other pixels with different spectra are regarded as the background. A corresponding bitmap recording these pixels will be recognized subsequently in the next stage, which will be discussed in the next section.

4. Proposed Block Recognition

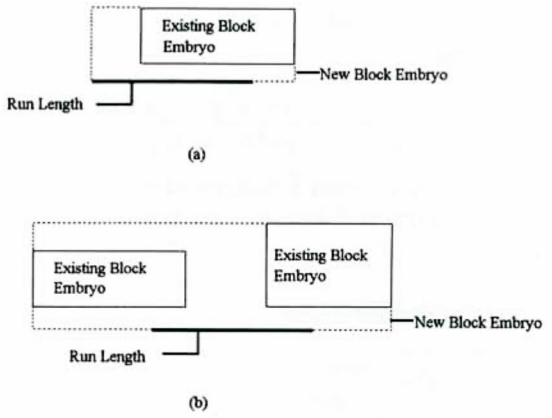


Figure 4. The process of block growing. (a) A block will grow if a run in the next scanning is adjacent to it; (b) two blocks are merged together if there exists a run adjacent to them.

4.1 Review of Common Features

A block which is extracted in the block segmentation stage may contain textlines or graphics. If the block type is identified in advance, unnecessary optical character recognition (OCR) procedures can be avoided. Without wasting time in OCR, the system will become more efficient.

Many features for discriminating texts and graphics proposed in the past are mostly obtained by statistical analysis. They are suitable for documents with regular and simple layouts, especially for those with isolated blocks, but are not all applicable to color documents with complicated backgrounds. More precise schemes for block recognition are necessary.

In our study, the block type recognition problem is transformed into the problem of recognizing textlines. This means that blocks which are rejected in textline recognition will be regarded as part of graphics. They are merged with other rejected blocks on each color plane to form graphics. The following features are frequently used in gray-scale document analysis. They are used in our system to decide if a block is a textline; if a block does not meet the constraint imposed by any of the features, it is rejected as a nontextline block.

(1) The height of a rectangular block:

$$H = \textit{block height}. \tag{2}$$

(2) The eccentricity of the surrounding rectangle of a block:

$$E = \frac{\textit{width of block}}{\textit{height of block}}. \tag{3}$$

If E is large, it implies the block is likely a textline.

(3) The saturation degree of a block:

$$S = \frac{\textit{number of block pixels}}{\textit{area of surrounding rectangle}}. \tag{4}$$

If S is close to 1, it reflects that the block has an approximately rectangular shape.

(4) The ratio of the transition number to the surrounding rectangle area:

$$T = \frac{total\ number\ of\ transitions}{width \cdot height}. \quad (5)$$

A transition occurs when a background pixel converts to a foreground pixel in each scanning. If T is large, the block may be a textline.

(5) The compact degree:

$$C = \frac{number\ of\ isolated\ pixels}{number\ of\ black\ pixels}. \quad (6)$$

Many graphic blocks are scattered with isolated pixels. On the contrary, a text block has few such pixels because texts are constituted with continuous lines. To compute C, for each black pixel, compute first the number of the neighbors with a 3×3 mask. If the number is smaller than 2, the pixel is regarded as isolated. Then the value of C can be computed using (6).

4.2 Improving Textline Recognition

Some properties other than the statistical features of texts will now be considered. It is evident that texts are constituted by slender lines. Based on this special property, we can make the assumption that lines constituting texts have almost identical widths, compared to graphics.

4.2.1 Computing average line width

We now describe the scheme we use for block recognition. It is helpful to define the line width at first. A line is constituted by small runs which are perpendicular to the line vector, as shown in Figure 5.

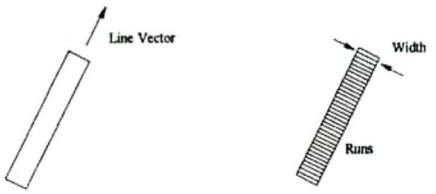


Figure 5. The line vector and the perpendicular constituting runs.

It is not practical, however, to compute the average width by these runs because the orientation of the lines are arbitrary and undetermined. An approximate measure technique using vertical and horizontal runs is employed before.

However, the result of computing the average horizontal or vertical runs to obtain the average width of the lines is not precise and might have large errors. Take the letter "E" as an example. If we compute the average length of the horizontal runs, the resulting value will be erroneous because the runs of the constituting lines vary greatly in length (see Figure 6(a)). The accurate average width is shown in Figure 6(b).

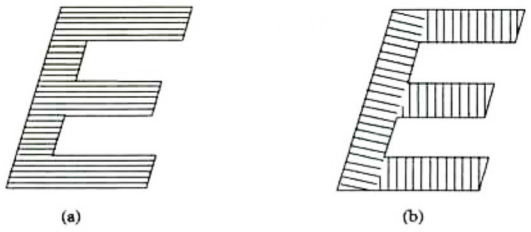


Figure 6. An example to compare erroneous approximate width with actual width. (a) erroneous approximate width computed with the original horizontal runs; (b) actual width computed from the runs perpendicular to the line vector.

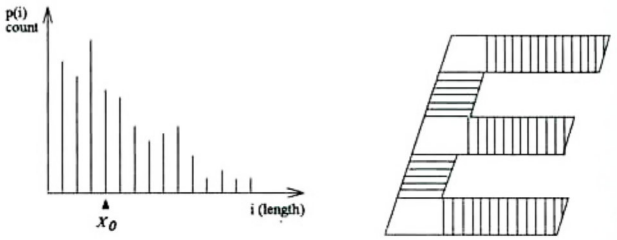


Figure 7. An example to illustrate the proposed line width detection method. (a) The histogram of the run lengths in both the horizontal and vertical directions; (b) the approximate width found by the proposed scheme.

An efficient measurement technique is used here. The histogram of the run lengths both in the horizontal and vertical directions is built at first. An example is shown in Figure 7(a). Because a line width is always shorter than a line length, the length of the shortest runs might be the line width. For this, we just find the value X_0 such that the sum of the run lengths over [0.. X_0] is equal to the number of the black pixels in the block. That is,

$$\sum_{i=0}^{X_0} P(i) \cdot i = C. \quad (7)$$

where C is the total number of the pixels and $P(\cdot)$ is the function shown by the histogram. The average line width can be obtained by the following formula:

$$avg\ line\ width = \frac{C}{\sum_{i=0}^{X_0} P(i)}. \quad (8)$$

The average line width so computed is more precise, as shown in Figure 7(b). The pixels in the three blank rectangles in the figure are ignored in the process. A new feature used in this study is then defined as:

$$L = \frac{C}{h \cdot \sum_{i=0}^{X_0} P(i)} \quad (9)$$

where h is the block height. By adding the block height to the denominator of the formula, the resulting value of L can be used to specify the relative scaling between the character height and the line width. It can be used to solve the problem of detecting textlines with different scalings; a block with its L value larger than a certain threshold is rejected as a non-textline.

4.2.2 Recognizing textlines

The average line widths or other features mentioned in the previous section carry only statistic information. They are necessary but not sufficient conditions, i.e., the non-text blocks may be classified as text blocks in some cases using these features. It is necessary to consider the relative position of pixels with respect to each other. A technique for recognizing textlines is introduced next.

First, scan a block vertically and horizontally and eliminate the runs whose lengths are far from the average line width. This results in two temporary bitmaps M_h and M_v. In order to recognize the textlines, labels have to be assigned to connected components.

Without loss of generality, take a horizontal scanning as an example. During scanning from left to right, the label 1 is assigned to the first run. The following runs adjoining the run labeled 1 in the next row are labeled 2. This process is continued until the last run of the connected component is reached. An example is shown in Figure 8(a).

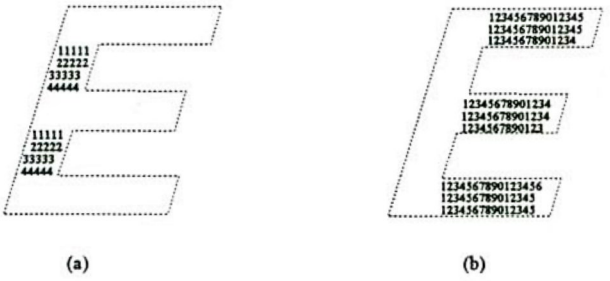


Figure 8. Results of textline detection in horizontal and vertical directions. (a) Labels of the proper runs in horizontal direction; (b) labels of the proper runs in horizontal direction. The lengths of these proper runs are close to the average width which is 4 pixels width in this object. LC = 0.8. If the threshold T is set to less than 0.8, it will be regarded as a text.

The largest label of each connected component is defined as the depth of the component. The component with a depth greater than a threshold T will be regarded as a line and kept; otherwise, the component will be removed. Then, perform a logical OR operation on the two bitmaps M_v and M_h, and a new bitmap like Figure 7(b) will be created.

Now, define the ratio

$$LC = \frac{pn(all\ lines\ detected)}{pn(block)} \qquad (10)$$

which can be utilized to check the text structure. The function $pn(x)$ means the pixel number of the object x. If LC is large, it means that the block has many lines and is a text block. By this technique, graphics can be discriminated more precisely.

The characters of some fonts may be constituted by lines with different widths such that the average width may be different from those of the constituting lines. Some errors should be ignored. The acceptable error range is proportional to the average run width. Table 1 is a mapping of average line widths to error ranges.

Table 1. The mapping of average line widths to error ranges.

Average line width	1	2	3	4	5	6	7	8	9	10	11
Acceptable error range	3	3	4	4	4	4	5	5	6	6	7

unit: pixel.

The line structure detection algorithm is described as follows. If all the feature values mentioned in Section 4.1 are valid, the line structure detection will be performed further.

Line Structure Detection Algorithm:

Step 1: Input block B.
Step 2: Compute average line width L.
Step 3: If L is invalid, then regard B as a partial graphic and exit.
Step 4: Record horizontal runs with lengths close to L to form a bitmap Mh.
Step 5: Record vertical runs with lengths close to L to form a bitmap Mv.
Step 6: Remove the components whose depth is less than a threshold in bitmap Mh and Mv.
Step 7: Combine Mh and Mv with the logical OR operation to form a bitmap M.
Step 8: Compute the pixel number in bitmap M. If the number is larger than a threshold, it implies that the block is full of lines and will be regarded as a textline.

5. Experimental Results

Several images were tested in our experiment. They were obtained by an HP ScanJet IIC color scanner at 250 dpi. The proposed color document segmentation method was tested on a Sun SPARC 10 workstation. The valid range for each feature is determined experimentally as shown in Table 2. They can be used to determine texts easily, but are not powerful enough to reject subareas of graphics. The improved approach proposed in the previous section can eliminate this problem.

Figure 9(a) is an image obtained from the Time magazine, which is a 24-bit 1861 × 2428 color image, and Figure 9(b) is the segmentation and block recognition result. The text blocks are nearly perfectly extracted. The segmentation time is about 165 seconds. Figure 9(c) shows the printout of the color plane which includes most of the texts in

Figure 9(a). Another document image in Figure 10(a) is taken from the MacUser magazine with size 1646 × 2096. The segmentation and block recognition result is shown as Figure 10(b). The segmentation process spent about 156 seconds. Figure 10(c) shows the printout of the color plane which includes most of the texts in Figure 18. Each gray rectangle in Figure 9(b) and Figure 10(b) means that the background is recognized as a graphic.

Table 2. The valid range for each feature, where E: eccentricity, S: saturation, T: transition number, C: compact degree, L: average line width.

Feature	E	S	T	C	L
Range	0.85 | ∞	0 | 0.6	0.05 | ∞	0 | 0.08	0 | 0.2

6. Conclusions

In this research work, image segmentation for color document analysis is studied. Similar to gray-scale document analysis, the system can be divided into several main components.

In our study, we take advantage of the color information sufficiently to extract the objects from a color document. By adaptive color quantization, millions of the colors are reduced into less than 250 colors. This tends to lower the complexity of analysis. For each color plane, the objects are extracted by the MCRLA which is extended from the CRLA. As a substitute for bitmap representation, the MCRLA use run-length representation, the time and space consuming problems for parallelism are thus solved.

As for block type recognition, a textline recognition approach is proposed. Statistical features often have trouble identifying a non-text object. The proposed approach can eliminate this drawback.

To sum up, it is convenient to extract the objects by color information. Even though the texts reside on graphics, they can be still identified exactly. Besides, the time and space consuming are also solved with the MCRLA in our system. Good experimental results have proved the feasibility of the proposed approach.

References

[1] K. Y. Wong, R. G. Casey, and F. M. Wahl, Document Analysis System, *IBM J. Res. Develop.*, submitted.

[2] G. Nagy, Preliminary investigation of techniques for automated reading of unformatted text, *Commun. ACM* 11, 1968, 480-487.

[3] E. G. Johnston, Printed text discrimination, *Computer Graphics and Image Processing* 3, 1974, 83-89.

[4] W. Scherl, F. Wahl, and H. Fuchsberger, Automatic separation of text, graphic and picture segments in printed material, In *Pattern Recognition in Practice* (E. S. Gelsema and L. N. Kanal, Eds., pp 213-221, North-Holland, Amsterdam, 1980.

[5] Friedrich M. Wahl. Kwan Y. Wong, and Richard G. Casey, "Block Segmentatioin and Text Extraction in Mixed Text/Image Documents," *Computer Graphics and Image Processing* **20**, 375-390, 1982.

[6] Norbert Bartneck, et al., "Document Analysis — From Pixels to Contents," *Proceedings of the IEEE*, Vol. 80, No. 7, July 1992, pp. 1101-1119.

[7] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, June 1992, 447-452.

[8] D. Wang and S.N. Srihari, "Classification of newspaper image blocks using texture analysis," *Comput. Vision Graph. Image Process.*, Vol. 47, 1989, pp. 327-352.

[9] K. Y. Wong, R. G. Casey and F. M. Wahl, "Document analysis system," *IBM J. Res. Devel.*, Vol. 26, No. 6, 1982, pp. 647-656.

[10] D. Wang and S. N. Srihari, "Classification of newspaper image blocks using texture analysis," *Computer Vision Graphics Image Process.*, 47, 1989, pp. 327-352.

[11] Teruo Akiyamaand Norihiro Hagita, "Automated Entry System for Printed Documents", *Pattern Recognition*, Vol. 23, No. 11, pp. 1141-1154, 1990.

[12] Ohuchi S., Imao K. and Yamada W., "A segmentation method for composite text/graphics (halftone and continuous tone photographs) documents," *Systems and Computers in Japan*, Vol. 24, 1993, 00. 35-44.

[13] Gloger J. M., "Use of the Hough transform to separate merged text/graphics in forms," *Proceeding. 11th IAPR International Conference on Pattern Recognition. Vol. 2, Conference B: Pattern Recognition Methodology and Systems*, Hague, Netherlands, August 1992, pp.268-271.

[14] Jain, A. K. and Bhattacharjee S., "Text segmentation using Gabor filters for automatic document processing," *Machine Vision and Applications*, Vol. 5, 1992, pp. 169-184.

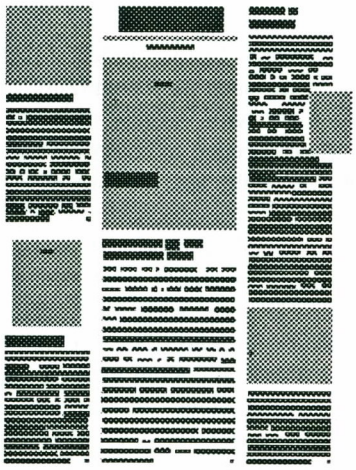
[15] Trupin E. and Lecourtier Y., "A modified contour following algorithm applied to document segmentation," *Proceeding. 11th IAPR International Conference on Pattern Recognition. Vol. 2, Conference B: Pattern Recognition Methodology and Systems*, Hague, Netherlands, August 1992, pp. 525-528.

[16] Gouin P., Scofield C. L., Gareyte C. and Pham H. "Neural network segmentation and recognition of text data on engineering documents," *IJCNN International Joint Conference on Neural Networks*, Baltimore, MD, USA, June 1992, Vol. 3, pp. 730-735.

[17] Shuichi Tsujimoto and Haruo Asada, "Major Components of a Complete Text Reading System", *Proceedings of the IEEE*, Vol. 80, No. 7, July 1992.

[18] Philippe Chauvet et al. "System for an intelligent office document analysis, recognition and description," *Signal Processing* 32, 1993, pp. 161-190.

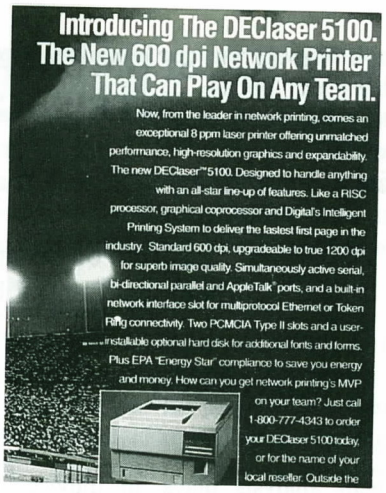
(a)



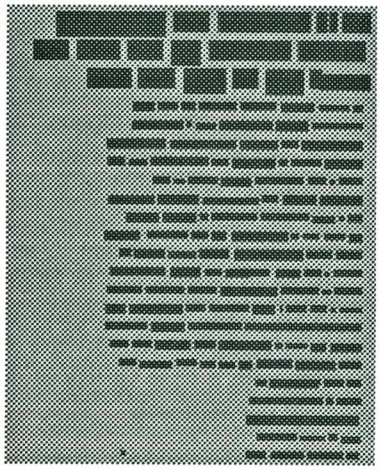
(b)

(c)

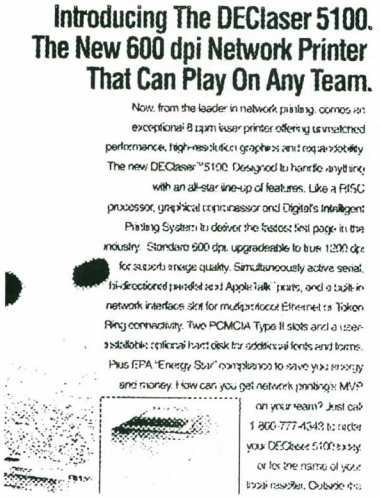
Figure 9. The Times document image and experimental results. (a) A tested color document image with size 1861×2428. (b) Result of segmentation and block recognition for (a). (c) Printout of the color plane which includes most of the texts in (a).



(a)



(b)



(c)

Figure 10. The MacUser document image and experimental results. (a) The original image. (b) Result of segmentation and block recognition for (a). (c) Printout of the color plane which includes most of the texts in (a).